****

**PROJECT REPORT ON**

**“ANDROID APPLICATION**

**ON SUPPLY CHAIN MANAGEMENT**

**IN OIL INDUSTRY”**

***FOR THE AWARD OF DEGREE***

*OF*

**BACHELOR OF ENGINEERING**

*IN*

**INFORMATION TECHNOLOGY**

***SUBMITTED BY:***

**MS. KANIKA DANKE (17)**

**MS. PRANOTI KULKARNI (35)**

**MS. RACHANA MAHALE (37)**

***UNDER THE GUIDANCE OF:***

**PROF. AADITYA DESAI**

(**PROFESSOR)**

ACADEMIC YEAR: **2014-2015**

**CERTIFICATE**

This is to certify that Kanika Sanjay Danke, Pranoti Raghavendra Kulkarni, Rachana Sadanand Mahale is a bonafide student of Thakur College of Engineering and Technology, Mumbai. They have satisfactorily completed the requirements of the PROJECT as prescribed by University of Mumbai while working on “Android Application on Supply Chain Management in oil industry” during academic year 2014-2015.

|  |  |  |
| --- | --- | --- |
| **Mr. Aaditya Desai**  **(Guide)** | **Dr. Vinayak Bharadi  (HOD IT)** | **Dr. B. K. Mishra**  **(Principal)** |
|  |  |  |
| **Internal Examiner External Examiner**  **(Name and Signature with Date) (Name and Signature with Date)**   |  | | --- | | Thakur College of Engineering and Technology  Kandivali (E), Mumbai-400101. |   PLACE: Mumbai  DATE: | | |

**PREFACE**

We take great opportunity to present this project report on “Android Application on Supply chain management in Oil Industry” and put before readers some useful information regarding our project.

We have made sincere attempts and taken every care to present this matter in precise and compact form, the language being as simple as possible.

We are sure that the information contained in this volume would certainly prove useful for better insight in the scope and dimension of this project in its true perspective.

The task of completion of the project though being difficulty was made quite simple, interesting and successful due to deep involvement and complete dedication of our group.

**Acknowledgements**

We take this opportunity to express our gratitude to everyone who helped us during the completion of our project. We are very grateful to all the faculty and friends who helped us by giving new ideas, techniques and various advices for implementing the project.

We would like to sincerely thank our project guide Mr. Aaditya Desai for his immense help and guidance through the project. His advice and ideas helped us ameliorate our project in many ways. His thorough interest in the project motivated us a lot. Thank you, Sir for your guidance.

Also we take this opportunity to thank Dr. Vinayak Bharadi, HOD IT Department and Dr. B.K. Mishra, Principal of Thakur College of Engineering and Technology, for allowing us to take up this project.

Last but never the least we thank all lab assistants for their cooperation in providing us the labs, required software and the network whenever needed. We also thank them to tolerate our disputes during the run.

Without the support acquired from all of you our project would rather be impossible. Thank you.

Signature and name of all the students in the group

**MS. KANIKA DANKE (17)**

**MS. PRANOTI KULKARNI (35)**

**MS. RACHANA MAHALE (37)**

**C O N T E N T S**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Chapter No. | Topic |  |  | Pg. No. |
|  | Certificate | |  | 2 |
|  | Preface | |  | 3 |
|  | Acknowledgement | |  | 4 |
|  | Abstract | |  | 6 |
| Chapter 1 | **Overview** | |  |  |
|  | 1.1 Importance of the project | |  | 7 |
|  | 1.2 Literature Survey | |  | 9 |
|  | 1.3 Scope of the Project | |  | 11 |
| Chapter 2 | **Proposed Work** | |  |  |
|  | 2.1 Problem Definition | |  | 12 |
|  | 2.2 Flow Chart & Data Flow Diagram | |  | 13 |
| Chapter 3 | **Analysis and Planning** | |  |  |
|  | 3.1 Feasibility Study | |  | 15 |
|  | 3.2 Project Planning | |  | 16 |
|  | 3.3 Scheduling | |  | 18 |
| Chapter 4 | **Design, Implementation and Installation** | |  |  |
|  | 4.1 Results Technology/ Software | |  | 22 |
|  | 4.2 Stage wise Model development | |  | 23 |
|  | 4.3 Installation stages | |  | 25 |
|  | 4.4 Implementation stages | |  | 32 |
|  | **Source Code** | |  |  |
|  | **Screenshots** | |  |  |
| Chapter 5 | **Result and Conclusion** | |  |  |
|  | References | |  |  |

**ABSTRACT**

Supply chain management (SCM) is the management of the flow of goods. It includes the movement and storage of [raw materials](http://en.wikipedia.org/wiki/Raw_material), work-in-process inventory, and finished goods from point of origin to point of consumption. Interconnected or interlinked networks, channels and node businesses are involved in the provision of [products](http://en.wikipedia.org/wiki/Product_(business)) and [services](http://en.wikipedia.org/wiki/Service_(economics)) required by end customers in a [supply chain](http://en.wikipedia.org/wiki/Supply_chain).[[2]](http://en.wikipedia.org/wiki/Supply_chain_management#cite_note-2) Supply chain management has been defined as the "design, planning, execution, control, and monitoring of supply chain activities with the objective of creating net value, building a competitive infrastructure, leveraging worldwide logistics, synchronizing supply with demand and measuring performance globally."SCM draws heavily from the areas of [operations management](http://en.wikipedia.org/wiki/Operations_management), [logistics](http://en.wikipedia.org/wiki/Logistics), [procurement](http://en.wikipedia.org/wiki/Procurement), and [information technology](http://en.wikipedia.org/wiki/Information_technology), and strives for an integrated approach. The thesis titled “Supply Chain Practices in Oil & Gas Industry” is aimed at studying the current supply chain practices, using the supply chain case studies of various Upstream and Downstream players. SCM strategy is discussed based on the literature review, analysis of the current practices and interaction with the practitioners. Firstly, the industry overview has been performed to establish a general understanding of the Oil & Gas Industry. Moreover, all factors influencing the value chain are observed. Analysis of the SC Practices will focus both – past and current practices. Secondly, case study analysis of main player(s) of the industry will be carried out operating in Pakistan. Based on the study, a discussion will be initiated to discuss the current issues and challenges. Possible solutions to optimize the chain would be concluded based on the research. Successful SCM requires a change from managing individual functions to integrating activities into key supply chain processes. In an example scenario, a purchasing department places orders as its requirements become known. The marketing department, responding to customer demand, communicates with several distributors and retailers as it attempts to determine ways to satisfy this demand. Information shared between supply chain partners can only be fully leveraged through [process integration](http://en.wikipedia.org/wiki/Process_integration). Supply chain business process integration involves collaborative work between buyers and suppliers, joint product development, common systems, and shared information. According to Lambert and Cooper (2000), operating an integrated supply chain requires a continuous information flow. However, in many companies, management has concluded that optimizing product flows cannot be accomplished without implementing a process approach. The key supply chain processes stated by Lambert (2004) are:

* Customer service management
* Demand management style
* [Order fulfillment](http://en.wikipedia.org/wiki/Order_fulfillment)

**Chapter 1**

**Overview**

* 1. **Importance of Project**

Smart phones and Tablets are more common than computers today. Almost everyone in the world make regular use of smart phones in their day to day lives. People can get a lot of different benefits from smart phones and that too in a very portable and mobile manner. Any kind of smart phone requires an operating system as an interface and the most popular OS today is Android. The android operating system presents never before seen flexibility and support for third-party applications. This has given rise to a huge amount of popularity of the Android OS not only among consumers, but also among developers. Android application development is fast becoming a separate field of information technology .More and more independent app developers and app development companies are taking interest in this OS and are coming up with some of the best apps around.

**The significance of the oil industry’s impact on the global economy is obvious. Oil supply chain management has to solve a lot of challenges caused by the nature of the supply chain in the oil industry such as complexity, in- flexible characteristics, long lead time, limited transportation forms at the different stages in the supply chain, rigid take or pay procurement and limited primary distribution capacity. Other challenges are caused by unforeseen events such as political or economic changes which have an impact on the price of the oil. This thesis seeks to add value by signifying and indicating optimization as a way to address uncertainties and points out a way to utilize resources efficiently in order to gain further development and cost savings in the long term. Finding options for optimization of the oil supply chain is vital because any cost saving means vast amounts of money for the oil companies therefore optimization is at the center of attention in the oil supply chain management.**





* 1. **Literature Survey**

The oil supply chain is also known to be a very complex chain compared to other industries[6]. It is due to several reasons. The whole oil supply chain is divided into up- and downstream segments based on activities before and after the refining stage. However, the distance from the oil exploitation point to the final consumers could often be thousands and thousands of kilometers (km) which is the main reason for the oil supply chain having longer lead time than in other industries. In addition, crude oil has to go through a complex, capital intensive refinery process as well[3]. The long lead time also indicates the involvement of various means of transport such as ships, pipelines, rail and road as well as high transportation cost.

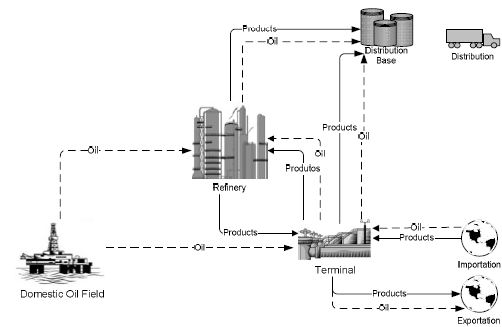


Figure: Typical Oil Supply Chain

(http://www.academia.edu/4660216/Final\_Proposal\_-\_Supply\_Chain\_Practices\_in\_Oil\_and\_Gas\_Industry\_by\_Waqas\_Ali\_Tunio)

## 

## Upstream

Upstream basically starts with the acquisition of crude oil and with the related operation such as exploration and production. Afterward logistics management has to be involved in order to deliver the crude oil from the exploitation point to the refinery.

1. *Exploration*: This stage involves seismic and geological operations.
2. *Production:* These concerns about exploitation of the crude oil from the reservoir by drilling. Production needs highly qualified engineering work and it also links to other activities such as procurement, transportation. The crude oil produced is transported by pipelines or oil tankers to the terminals for storage. From here it is either to transport directly to the refinery or exported to other companies’ refineries[5].

## Downstream

The first stage is the refining process which is based on demand forecasting and triggers the procurement and the logistics activities in order to supply crude oil to the refinery and deliver its derivatives to the customers.

1. *Procurement:* This is sourcing of and managing the supply of the raw material to the refinery in the right time and in the right quantity.
2. *Refining:* This is a complex, well planned process which involves the transformation of the crude oil into different types of derivatives based on demand forecasting. Therefore, this has a tight link to the next stage, to the marketing activities and also involves inventory management.
3. *Distribution:* Logistics management assures that the right products get delivered to the right customers in the right time preferably in a cost efficient way[4].
4. *Marketing:* This deals with marketing the different crude oil derivatives to the right customers. Marketing has to have an accurate knowledge about the current inventory level and refinery activities in order to manage its sale function.

**1.3 Scope of project**

**There was a time when mobile phones were developed only for voice communication. But eventually with the advancement in technology, mobile phones have turned out to be smart phones indeed. There are much more added functionalities which are of major focus of interest into this device. Two such major factors are the internet and GPS services. Both of these functionalities are already implemented but are only in the hands of manufacturers not in the hands of users because of proprietary issues, the system does not allow the user to access the mobile hardware directly. But now, after the release of android based open source mobile phone a user can access the hardware directly and design customized applications to develop internet and GPS enabled services and can also program the other hardware components like camera etc.** Supply Chain Management of Oil & Gas industry is of the most volatile & challenging nature. Hence, a proper study is required which can be used as a reference to know about the past and current practices of the country in this sector along with the issues faced, in order to be able to devise more options when planning for future.

Building an android application on Supply chain management for oil industry will reduce the nature of complexity which persists between the supplier and the customer since ultimately “Customer is a King”. The transportation cost, lead time, everything should be handled appropriately to eliminate any inflexibilities in this chain. The customer should get timely access to all the resources he needs. Hence, this app is expected to fulfil those requirements.

**Chapter 2**

**Proposed Work**

**2.1. Problem definition**

However, every coin has two sides. There is no doubt that the scope of the project is humongous but along with it various questions are raised on the availability, usability, reliability, efficiency, maintainability, portability, testability and flexibility of application. The availability problem states that to whom this application would be available.

The main focus of application would be on customer as this is supply chain related project the end user has to be satisfied with all the services he is getting. A recent survey done by us on this area is that the supply chain itself is a complex and intricate sector so customer satisfaction is usually very less. Hence we will cater to their needs. Although our focus will be on customer needs but we have also added certain functionalities that will help others in this supply chain as well. Thus the usability problem also has a solution.

The next problem which can surface is the efficiency of application. The question of how efficient is the application can be asked by the user. Android is an open platform technology and allow you more control and customization. Hence it is recommended by the professionals in the IT world. Here is a survey summarizing the fact that android is recommended over other technologies.

Next the problem that can surface is the maintainability of the application. The maintainability can be kept in check by providing the user with newer versions of the application if any bugs are found in the current one. Portability is another area in which the user can be concerned. But since Android is an open platform technology, the application’s induction into the Android Market will handle the portability concern too. The category of the user that uses this application spans a vast diversity. Lastly the area of concern would be related to the flexibility of application. But since we are motivated cater to the different version of Android viz. Android 2.2 (Froyo) and Android 2.3 (Gingerbread), this concern shouldn’t arise. Thus major domain in which any problem could arise has been covered.

**2.2 FLOWCHART:**

**A**

**Take User Input**

**Select Operation**

**Show Home Page**

**YES**

**NO**

**Input User details**

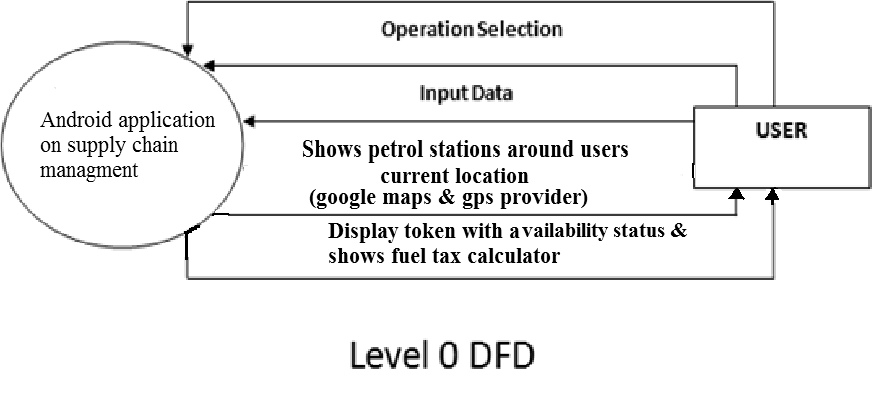
**Store detail in Database**

**Is mobile no. or password valid?**

**Load Activity**

**Start**

**Data Flow Diagram:**

****

**Store**

**Enters**

**Show**

**User Database**

**USER**

**Level 1: DFD**

**Chapter 3**

**Analysis and Planning**

**3.1 Feasibility Study**

Cost factor in implementing this mobile application is only limited to information gathering, and managing the server database. The feasibility study based on various factors is as follows:

**3.1.1 Technical and system feasibility:**

Android OS will be used as the platform on which this application works, Android OS is open source software. Thus, the cost for acquisition of this software is eliminated. The only cost incurred is of maintaining the Software, which is also limited due to open source nature of it and updating the database.

Open source software is easier to maintain compared to paid software, because we do not need to buy them again and also because we can find its solution on various forums.

**3.1.2 Schedule feasibility:**  Schedule feasibility is a very important dimension of the Project as all the databases need to be up to date. The application must show the user the various operations that can be performed and actions to be taken thereafter.

**3.1.3 Market feasibility:** The focus is on a single user. The application caters to the need of a single user initially and then it can be used by multiple users.

**3.1.4 Financial feasibility:** Financial capital investment includes the cost required to make an account on the Android market from where the users will be able to download this Application for free.

**3.1.5 Resource feasibility:**

Eclipse IDE was used to make this application along with SQLite database that android phones use. Eclipse is a standard tool used for Android Application development due to its user friendliness in providing simple drag and drop Facilities and an emulator that works just like any other android phone.

**3.2 Project Planning:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Duration | Start | Finish |
| 1.Requirement Analysis | **25 days** | **2/08/2014** | **27/08/2014** |
| 1.1 Software specification | 4 days | 2/08/2014 | 6/08/2014 |
| 1.2 Presentation | 7 days | 6/08/2014 | 13/08/2014 |
| 1.3 In-house requirement specification | 2 days | 13/08/2014 | 15/08/2014 |
| 1.4 SRS | 8 days | 15/08/2014 | 23/08/2014 |
| 1.5 Requirement Gathering | 4 days | 23/08/2014 | 27/08/2014 |
| 2.Analysis | **12 days** | **27/08/2014** | **8/09/2013** |
| 2.1 User Requirements | 3 days | 27/08/2014 | 30/08/2014 |
| 2.2 Functional Requirements | 5 days | 30/08/2014 | 4/09/2014 |
| 2.2 Non functional Requirements | 4 days | 4/09/2014 | 8/09/2014 |
| 3. Design | **21 days** | **8/09/2014** | **29/09/2014** |
| 3.1 Architecture Design | 6 days | 8/09/2014 | 14/09/2014 |
| 3.2 Database Schema | **7 days** | 14/09/2014 | 21/09/2014 |
| 3.3 Graphical User Interface | 8 days | 18/09/2014 | 29/09/2014 |

**Table 3.3: Semester VII Timeline**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Duration | Start | Finish |
| 4. Coding /Implementation | **60 days** | **25/01/2015** | **25/03/2015** |
| 4.1 Database Creation | 3 days | 25/01/2015 | 28/01/2015 |
| 4.2 Software Development | 12 days | 28/01/2015 | 10/02/2015 |
| 4.3 Database Integration | 4 days | 10/02/2015 | 14/02/2015 |
| 4.5 Coding and Implementation | 20 days | 14/02/2015 | 6/03/2015 |
| 4.4 Integration | 6 days | 6/03/2015 | 12/03/2015 |
| 4.5 Implementation of Application | 20 days | 12/03/2015 | 25/03/2015 |
| 5. Verification and Testing | **30 days** | **25/03/2015** | **25/04/2015** |
| 5.1 Unit Testing | 5 days | 25/03/2015 | 30/03/2015 |
| 5.2 Stress Testing | 5 days | 30/03/2015 | 05/04/2015 |
| 5.3 Alpha/Beta Testing | 6 days | 5/04/2015 | 11/04/2015 |
| 5.4 Acceptance testing | 5 days | 11/04/2015 | 16/04/2015 |
| 5.5 Performance Testing | 5 days | 16/04/2014 | 21/04/2015 |
| 5.6 Modification | 4 days | 21/04/2014 | 25/04/2015 |

**Table 3.4: Semester VIII Timeline**

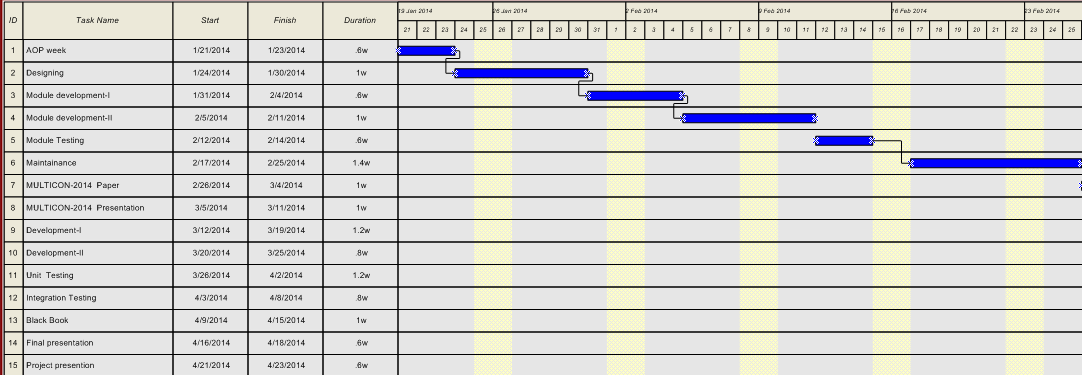
**3.3. Scheduling**

**Gantt chart:**

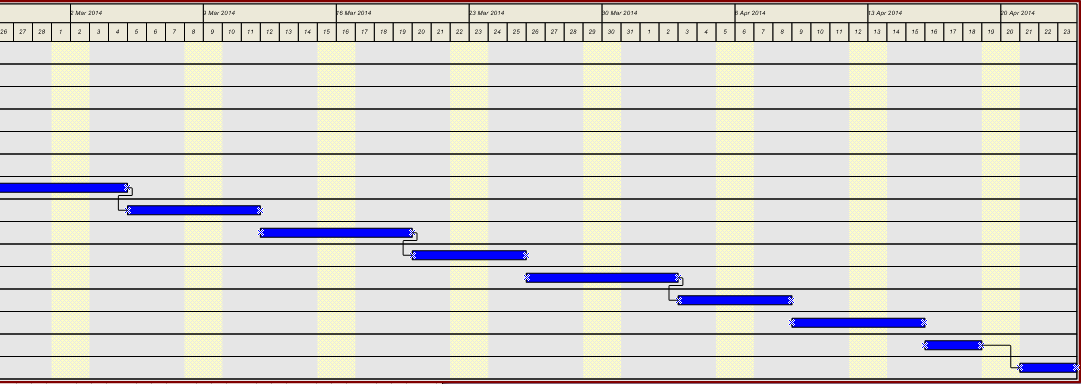
A Gantt chart is a type of [bar chart](http://en.wikipedia.org/wiki/Bar_chart), developed by [Henry Gantt](http://en.wikipedia.org/wiki/Henry_Gantt) in the 1910s, that illustrates a [project schedule](http://en.wikipedia.org/wiki/Schedule_(project_management)). Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a [project](http://en.wikipedia.org/wiki/Project).

Terminal elements and summary elements comprise the [work breakdown structure](http://en.wikipedia.org/wiki/Work_breakdown_structure) of the project. Some Gantt charts also show the [dependency](http://en.wikipedia.org/wiki/Dependency_(project_management)) (i.e. precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.

Although now regarded as a common charting technique, Gantt charts were considered revolutionary when first introduced.[[1]](http://en.wikipedia.org/wiki/Gantt_chart#cite_note-1) In recognition of [Henry Gantt](http://en.wikipedia.org/wiki/Henry_Gantt)'s contributions, the [Henry Laurence Gantt Medal](http://en.wikipedia.org/wiki/Henry_Laurence_Gantt_Medal) is awarded for distinguished achievement in management and in community service. This chart is also used in [information technology](http://en.wikipedia.org/wiki/Information_technology) to represent data that has been collected.



**Figure 3.5: Gantt chart of Project development [Jan-Feb]**



**Figure 3.6: Gantt chart of Project development [Mar-Apr]**

**Advantages:**

* **It creates a picture of complexity.** I am quite a fan of diagrams and charts. We think in pictures. Therefore, if we can see complex ideas as a picture, this will help our understanding.
* **It organizes your thoughts.** I am also a big fan of the concept of dividing and conquering. A big problem is conquered by dividing it into component parts. A Gantt chart will force you to do this.
* **It demonstrates that you know what you’re doing.** When you produce a nicely presented Gantt chart with high level tasks properly organized and resources allocated to those tasks, it speaks volumes about whether you are on top of the needs of the project and whether the project will be successful.
* **It (should) help you to set realistic time frames.** The bars on the chart indicate in which period a particular task or set of tasks will be completed. This can help you to get things in perspective properly.
* **It can be highly visible.** It can be useful to place the chart, or a large version of it, where everyone can see it. This helps to remind people of the objectives and when certain things are going to happen.

**Chapter 4**

**4.1 Results Technology / Software:**

****

**Figure 4.1.1: Activity Diagram**

**4.2 Stage wise Model development, Flow Chart**

Software Development Model Used:

Software process model deals with the model which we are going to use for the development of the project. There are many software process models available but while choosing it we should choose it according to the project size that is whether it is industry scale project or big scale project or medium scale project.

Accordingly the model which we choose should be suitable for the project as the software process model changes the cost of the project also changes because the steps in each software process model varies.

This software is build using the waterfall mode. This model suggests work cascading from step to step like a series of waterfalls. It consists of the following steps in the following manner

**Incremental model:**

**Analysis Phase**

**Design Phase**

**Coding Phase**

**Testing Phase**



**Figure 4.1.2: Incremental Model**

**Following are the different phases of Incremental Model:**

**Analysis Phase:**

To attack a problem by breaking it into sub-problems. The objective of analysis is to determine exactly what must be done to solve the problem. Typically, the system’s logical elements (its boundaries, processes, and data) are defined during analysis.

**Design Phase:**

The objective of design is to determine how the problem will be solved. During design the analyst’s focus shifts from the logical to the physical. Data elements are grouped to form physical data structures, screens, reports, files and databases.

**Coding Phase:**

The system is created during this phase. Programs are coded, debugged, documented, and tested. New hardware is selected and ordered. Procedures are written and tested. End-user documentation is prepared. Databases and files are initialized. Users are trained.

**Testing Phase:**

Once the system is developed, it is tested to ensure that it does what it was designed to do. After the system passes its final test and any remaining problems are corrected, the system is implemented and released to the user.

**4.3 Installation Stage**

Here’s a quick start installation guide to get you up and running with the Google Android Software Development Kit (SDK). This guide will describe how to install the Android SDK and set up your chosen development environments. If you’ haven’t already done so you can download the Android SDK from the link below, then we can get started.

## System Requirements

In order to first use the Android SDK code and tools for development you will of course need a suitable environment develop from.

Currently the following operating systems are supported:

* Windows XP or Vista
* Mac OS X 10.4.8 or later (x86 only)
* Linux (tested on Linux Ubuntu Dapper Drake)

You will also need to install a suitable development environment such as:

* Eclipse

[**Eclipse**](http://www.eclipse.org/downloads/) Juno  
[**Android Development Tools plugin**](http://code.google.com/android/intro/installing.html#installingplugin) (optional)

* Other development environments or IDEs

[**JDK 5 or JDK 6**](http://java.sun.com/javase/downloads/index.jsp) (JRE alone is not sufficient)  
Not compatible with Gnu Compiler for Java (gcj)  
[**Apache Ant**](http://ant.apache.org/) 1.6.5 or later for Linux and Mac, 1.7 or later for Windows

## Installing the Android SDK

First you will need to download the Android SDK pack .zip archive, once downloaded find a suitable installation location on your machine and extract the zipped files.

Please note: This installation location will be referred to as $SDK\_ROOT from now on through this tutorial

Alternatively you can add /tools to your root path which will prevent the need to specify the full path to the tools directory along with enabling you to run Android Debug Bridge (adb) along with other command line tools.

### Linux

1. Edit the ~/.bash\_profile or ~/.bashrc files looking for a line that sets the PATH variable.
2. Add the full path location to your $SDK\_ROOT/tools location for the PATH variable.
3. If no PATH line exists you can add the line by typing the following:
4. export PATH=${PATH}:<path to your $SDK\_ROOT/tools>

### Mac OS X

1. In the home directory locate the .bash\_profile and locating the PATH variable add the location to your $SDK\_ROOT/tools folder.

### Windows XP / Vista

1. Right click on the My Computer icon and select the properties tab.
2. Select the Advanced tab and click the Environment Variables button.
3. In the new dialog box double-click on Path (located under System Variables) and type in the full path location to the tools directory.

The Android SDK also requires a suitable development environment to work in, here’s the installation guides for each of the supported environments.

## Android Eclipse Plug-in (ADT)

If you choose to use the Eclipse IDE as your Android development environment you will have the opportunity to install and run a plug-in called Android Development Tools. ADT comes with a variety of powerful tools and extensions that will make creating, running and debugging your Android applications much easier and faster.

In order to download and install ADT you will first need to configure an Eclipse remote update, this can achieved via the following steps:

1. Start Eclipse, then select **Help** > **Software Updates** > **Find and Install….**
2. In the dialog that appears, select **Search for new features to install** and press **Next**.
3. Press **New Remote Site**.
4. In the resulting dialog box, enter a name for the remote site (e.g. Android Plug-in) and enter this as its URL: https://dl-ssl.google.com/android/eclipse/.
5. Press **OK**.
6. You should now see the new site added to the search list (and checked).
7. Press **Finish**.
8. In the subsequent Search Results dialog box, select the checkbox for **Android Plug-in** >**Eclipse Integration** > **Android Development Tools** and press **Next**.
9. Read the license agreement and then select **Accept terms of the license agreement**, if appropriate.
10. Press **Next**.
11. Press **Finish**.
12. The ADT plug-in is not signed; you can accept the installation anyway by pressing **Install All**.
13. Restart Eclipse.
14. After restart, update your Eclipse preferences to point to the SDK root directory ($SDK\_ROOT):   
    Select **Window** > **Preferences**… to open the Preferences panel. (Mac OS X: **Eclipse** >**Preferences**)  
    Select **Android** from the left panel.  
    For the SDK Location in the main panel, press **Browse.**.. and find the SDK root directory.
15. Press **Apply**, then **OK**

## How-To Use Eclipse to Develop Android Applications

In order to begin development on your Android applications you will first need to create a new Android project and then configure a launch configuration. Once completed you will have the capability to write, run and debug your Android creations.

The following sections below will provide you with the necessary instructions to get you up and running with Android provided you have installed the ADT plug-in (as previously mentioned) in your Eclipse environment.

### Creating a New Android Project

The Android Development Tools plug-in kindly provides a Wizard for setting up new Projects which will allow us to create new Eclipse projects relatively quickly for either new or existing code.

Select File > New > Project

1. Select Android > Android Project, and press Next
2. Select the contents for the project:

* Select Create new project in workspace to start a project for new code. Enter the project name, the base package name, the name of a single Activity class to create as a stub .java file, and a name to use for your application.
* Select Create project from existing source to start a project from existing code. Use this option if you want to build and run any of the sample applications included with the SDK. The sample applications are located in the samples/ directory in the SDK. Browse to the directory containing the existing source code and click OK. If the directory contains a valid Android manifest file, the ADT plug-in fills in the package, activity, and application names for you.

Press **Finish**.

Once completed the ADT plug-in will go ahead and create the following files and folders as appropriate for the type of project selected:

* src/ A folder that includes your stub .java Activity file.
* res/ A folder for your resources.
* AndroidManifest.xml The manifest for your project.

### Creating a Launch Configuration for Eclipse

In order to be able to run and debug your own Eclipse applications you must first create a launch configuration. Simply, a launch configuration is used to specify which project to launch, which activity to start and the specific emulation options to use.

To create a launch configuration for the application, please see the following steps:   
1. Select **Run** > **Open Run Dialog**… or **Run** > **Open Debug Dialog**… as appropriate.  
2. in the project type list on the left, right-click **Android Application** and select **New**.  
3. Enter a name for your configuration.  
4. On the Android tab, browse for the project and Activity to start.  
5. On the Emulator tab, set the desired screen and network properties, as well as any other emulator start-up options.  
6. You can set additional options on the Common tab as desired.  
7. Press **Apply** to save the launch configuration, or press **Run** or **Debug** (as appropriate).

### Running and Debugging an Eclipse Application

Once both steps 1 and 2 have been completed and your project and launch configurations are up and running you will now be able to run or debug your application.

From the Eclipse main menu, select **Run** > **Run** or **Run** > **Debug** as appropriate. This command will run or debug the most recently selected application.

To set or change the active launch configuration, use the Run configuration manager, which you can access through **Run** > **Open Run Dialog**… or **Run** > **Open Debug Dialog**….

Running or debugging the application will trigger the following actions:

* Starts the emulator, if it is not already running.
* Compile the project, if there have been changes since the last build, and installs the application on the emulator.
* Run starts the application.
* Debug starts the application in “Wait for debugger” mode, then opens the Debug perspective and attaches the Eclipse Java debugger to the application.

### Developing Android Applications with Other IDEs and Tools

Although it is recommended you use Eclipse with the Android plug-in to develop your applications, the SDK also provides tools which will enable you to develop with other IDE’s including intelliJ (alternatively you could just use Eclipse without the plug-in).

### How-To Build an Android Application

Here’s how to use the Ant build.xml file generated by activityCreator to build your application.

1. If you don’t have it, you can obtain Ant from the Apache Ant home page. Install it and make sure it is on your executable path.
2. Before calling Ant, you need to declare the JAVA\_HOME environment variable to specify the path to where the JDK is installed. Note: When installing JDK on Windows, the default is to install in the “Program Files” directory. This location will cause ant to fail, because of the space. To fix the problem, you can specify the JAVA\_HOME variable like this: set JAVA\_HOME=c: \Prora~1\Java\. The easiest solution, however, is to install JDK in a non-space directory, for example: c:\java\jdk1.6.0\_02.
3. If you have not done so already, follow the instructions for creating a New Project above to set up the project.
4. You can now run the Ant build file by simply typing ant in the same folder as the build.xml file for your project. Each time you change a source file or resource, you should run ant again and it will package up the latest version of the application for you to deploy.

### How-To Run an Android Application

In order to run a compiled application you will first need to upload the .apk file to the /data/app/ directory in the emulator using the adb tool:

1. Start the emulator (run $SDK\_HOME/tools/emulator from the command line)
2. On the emulator, navigate to the home screen (it is best not to have that application running when you reinstall it on the emulator; press the Home key to navigate away from that application).
3. Run adb install myproject/bin/<appname>.apk to upload the executable. So, for example, to install the Lunar Lander sample, navigate in the command line to $SDK\_ROOT/sample/LunarLander and type.../../tools/adb install bin/LunarLander.apk
4. In the emulator, open the list of available applications, and scroll down to select and start your application.

### How-To Attach a Debugger to Your Application

The following section details how to display debug information directly onto the screen (for example CPU usage). It also shows you how to hook up your IDE to debug running applications on the emulator.

**Start** the [**Dalvik Debug Monitor Server (DDMS)**](http://code.google.com/android/reference/ddms.html) tool, which acts as a port forwarding service between your IDE and the emulator.

1. Set optional debugging configurations on your emulator, such as blocking application startup for an activity until a debugger is attached. Note that many of these debugging options can be used without DDMS, such as displaying CPU usage or screen refresh rate on the emulator.
2. Configure your IDE to attach to port 8700 for debugging. We’ve included information higher up on how to set up Eclipse to debug your project.

### Debugging Android

Google Android has a fairly extensive set of tools to help you debug your programs:

* [**DDMS**](http://code.google.com/android/reference/ddms.html) – A graphical program that supports port forwarding (so you can set up breakpoints in your code in your IDE), screen captures on the emulator, thread and stack information, and many other features. You can also run logcat to retrieve your Log messages. See the linked topic for more information.
* [**logcat**](http://code.google.com/android/reference/ddms.html#logcat) – Dumps a log of system messages. The messages include a stack trace when the emulator throws an error, as well as Log messages. To run logcat, see the linked topic. …  
  I/MemoryDealer( 763): MemoryDealer (this=0x54bda0): Creating 2621440 bytes heap at 0x438db000  
  I/Logger( 1858): getView() requesting item number 0  
  I/Logger( 1858): getView() requesting item number 1  
  I/Logger( 1858): getView() requesting item number 2  
  D/ActivityManager( 763): Stopping: HistoryRecord{409dbb20com.google.android.home.AllApps}  
  …
* [**Android Log**](http://code.google.com/android/reference/android/util/Log.html)- A logging class to print out messages to a log file on the emulator. You can read messages in real time if you run logcat on DDMS (covered next). Add a few logging method calls to your code.

To use the Log class, you just call Log.v() (verbose), Log.d() (debug), Log.i() (information), Log.w() (warning) or Log.e (error) depending on the importance you wish to assign the log message.  
Log.i(“MyActivity”, “MyClass.getView() — Requesting item number ” + position) You can use logcat to read these messages

* [**Eclipse plug-in**](http://code.google.com/android/intro/installing.html#developingwitheclipse) – The Eclipse Android plug-in incorporates a number of these tools (ADB, DDMS, logcat output, and other functionality). See the linked topic for more information.

Debug and Test Device Settings – Android exposes several settings that expose useful information such as CPU usage and frame rate.

**4.4 Implementation stages**

The National Implementation Research Network (NIRN) defines implementation as “a specific set of activities designed to put into practice an activity or program” (Fixsen, Naoom, Blase, Friedman, & Wallace, 2005, p. 5). NIRN has researched program implementation across disciplines, including social services, business, engineering, and education, providing a broad overview of the challenges and facilitators. Its review highlights major problems in implementation practice, such as relying solely on implementation “by edict” or training alone, or implementing a new intervention without fidelity, without a broad enough scale to effect change, or without a plan for sustainability.

# Stages of Implementation:

NIRN's work is based on a cross-disciplinary literature review, which identified a sequence of stages that implementation efforts must address in order to be successful: *exploration and adoption*, *program installation*, *initial implementation*, *full implementation*, *innovation*, and *sustainability*. These stages represent an iterative process, as efforts are reassessed or revaluated in light of new realities.

NIRN estimates that the first four phases described here take systems from two to six years to implement. In addition, NIRN emphasizes the importance of aligning the goals of the immediate setting with the layers of settings that connect it to the larger community—the district, the community, and the state. Starting with the goal in mind is essential to implementation efforts’ integrity and evaluation plans. Planning for a clear goal will help chart a particular course with specific choices, consequences, and intended outcomes. NIRN’s work highlights the importance of a dedicated implementation team – a single champion cannot carry out the system-level work necessary to effect real school-wide change.

## Exploration and Adoption:

This stage is the initial process of problem articulation and solution identification. What is the problem that requires focused remediation? For example, “reducing bullying” is too vague; “reducing the number of incidents of harassment and bullying on afternoon bus rides from the high school" is a much more narrowly identified problem.

Getting to the heart of an identified problem is not easy. Many layers of perceptions, assumptions, and habits often have to be peeled away. It may require several months of data analysis and discussion to pinpoint areas of systemic weakness.

Bringing practitioners and stakeholders together around the solution is a challenge that needs to be faced in this early stage. The implementation teams need to have a dialogue with colleagues and the larger community to ensure that the problem and solution are aligned with the broad goals of the community.

## Program Installation:

This stage focuses on the system that is being altered in order to take on the process of implementing a new program or solution. Dean Fixsen, co-director of NIRN, shares that this stage is the most often overlooked in education. Schools and practitioners, it seems, do not have the organizational habit as do community-based organizations or commercial enterprises that an existing system needs to be “built out” in advance of starting a new program. Rather, schools often adopt a new program or initiative as an add-on to existing staff, time, equipment, and commitments. Too often, the result is a disappointment for all involved.

Ideally, this stage of implementation involves taking stock of existing resources—human, physical, and financial—for possible reassignment, as well as addressing resource gaps for the planned new program. Some questions to ask at this stage include:

* Do new policies need to be written to reflect the upcoming changes?
* Do existing staff members have the expertise to implement the identified solution or are new hires necessary?
* Who will deliver the training, and when and where? How will staff and stakeholders be paid or compensated for attending the training?
* How will the effectiveness of the training be measured and how will ongoing learning be supported?
* Are our physical space, infrastructure, and equipment adequate?
* Can additional funding streams be sought to cover the long-term costs?
* What outcome measures will be watched and how will progress be tracked? What are the benchmarks? What metrics will be used?

## Initial Implementation

While program installation focuses on the alterations to the system, initial implementation focuses on the changes that must occur in practice. Resistance may arise when practitioners experience the uncomfortable sensation of changing their practice—using new language, routines, or documentation. Supporting practitioners through this stage is critical, NIRN cautions, “when the program is struggling to begin and when confidence in the decision to adopt the program is being tested” (Fixsen et al., 2005, p. 16). This is not a pilot test of the program, but the initial roll-out with practitioners.

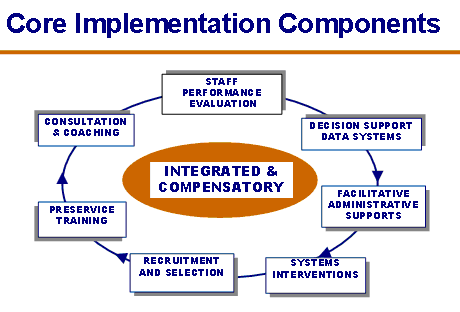
Schools and districts can get stuck in this mode of initial implementation, trying on one solution after another, but not persisting through the initial resistance until practitioners incorporate the program into their practice. If coupled with a lack of attention to preparing the system to incorporate a new intervention in the program installation phase, the intervention will enter the system on very shaky ground indeed; it is not a surprise in these circumstances that practitioners resist rather than try to accommodate a new program without additional support.

Factors found to be key supports to program implementation include:

* professional development that is planned, intentional, and part of an ongoing, school-wide effort;
* leadership, in which school leaders serve as role models, cheerleaders, and facilitators, and ensure that teachers have the needed resources;
* organization and a structure that supports, encourages and recognizes change efforts; and
* resources and support, internal and external, such as administrative leadership, buy-in from the community, and partnerships with other relevant organizations.

## Full Implementation:

This phase is sensitive to the balances in the system that got it started. As staff shift—in roles and responsibilities as well as in number—the core team of champions is likely to change, meaning the original training grows diffuse among the team. Change in leadership roles, in particular, expose the implementation process to setbacks and detours. NIRN’s model of Core Implementation Components (see figure below) illustrates an understanding of how these inevitable shifts can be counterbalanced with planning and a focus on program-centred practice, rather than practitioner-centred practice.



**Figure 4.2: Core Implementation Component**

Systems in this phase of implementation are ready to research the effectiveness of their efforts and revisit alignment with the broader goals of the community, which may have shifted since the program was developed. Those involved might work with the program developers or others to determine the effectiveness of a program functioning in place compared to the evidence base behind the program. Questions to consider include:

* Are the outcome measures at the expected level of performance? Why or why not?
* Are the expected performance standards still appropriate?
* Have the goals of the community and district shifted since the program began? Is the program still aligned to the larger goals?
* What does a cost analysis of training and investment show as a return on investment with this program?
* What do our practitioners think of this program? Has it become expected practice and spread beyond the champions?

These types of questions go beyond annual evaluation data to a more in-depth evaluation of the process from the inception.

## Innovation

Sites and teams should attempt to implement a program with as much fidelity as possible from the early stages in order to collect and evaluate outcome data and compare results against the evidence base. This will help establish the credibility of the implementation effort and validate the hard work practitioners invest in implementing a new program. How can your team identify those elements that characterize a program and are key to the successful outcomes reported in the literature? What can change and what must be adopted as is? These questions are real challenges for practitioners far removed from the original researchers and program designers. One solution is to engage the designers or researchers in dialogue or consultation about your sites’ adaptations.

## Sustainability

Planning with the ends in mind means that sustainability, far from the last stage on the list, is an integral part of the implementation process as a whole. We have all seen implementation efforts led by a single champion or team that have struggled and withered when one member of the team is reassigned or the external coaching support is withdrawn. Long-term vision should be an integral part of the entire implementation model, requiring that teams focus on support, scaling up, and sustainability from the inception of the planning process.

Bottom of Form

**SOURCE CODE**

**1.Login.java**

package com.example.PetrolFinder;

import com.example.PetrolFinder.R;

import android.app.Activity;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.text.InputType;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.Toast;

public class Login extends Activity{

Intent i=null;

ImageView im=null;

EditText tv1,tv4;

boolean flag=false;

SQLiteDatabase db=null;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.login);

im=(ImageView)findViewById(R.id.show\_hide2);

tv1=(EditText)findViewById(R.id.phone2);

tv4=(EditText)findViewById(R.id.password2);

db=openOrCreateDatabase("mydb", MODE\_PRIVATE, null);

// db.execSQL("create table if not exists login(name varchar,mobile\_no varchar,email\_id varchar,password varchar,flag varchar)");

im.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View arg0) {

if(flag==false)

{

im.setImageResource(R.drawable.hide);

tv4.setInputType(InputType.TYPE\_TEXT\_VARIATION\_PASSWORD);

flag=true;

}

else

{

im.setImageResource(R.drawable.show);

tv4.setInputType(129);

flag=false;

}

}

});

}

public void action(View v)

{

switch(v.getId())

{

case R.id.signin2:

i=new Intent(this,Signin.class);

startActivityForResult(i, 500);

overridePendingTransition(R.anim.slide\_in\_top, R.anim.slide\_out\_bottom);

finish();

break;

case R.id.start:

String mobile\_no=tv1.getText().toString();

String password=tv4.getText().toString();

if(mobile\_no==null||mobile\_no==""||mobile\_no.length()<10)

{

show("Please Enter Correct mobile number.");

}

else if(password==null||password==""||password.length()<6)

{

show("Please Enter Correct Password.");

}

else

{

Cursor c=db.rawQuery("select \* from login where mobile\_no='"+mobile\_no+"' and password='"+password+"'",null);

c.moveToFirst();

if(c.getCount()>0)

{

i=new Intent(this,Welcome.class);

startActivityForResult(i,500);

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);

db.close();

finish();

}

else

show("Wrong Password or Mobile number.");

}

break;

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

}

public void show(String str)

{

Toast.makeText(this, str, Toast.LENGTH\_LONG).show();

}

}

**2.Register.java**

package com.example.PetrolFinder;

import com.example.PetrolFinder.R;

import android.app.Activity;

import android.content.Intent;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.text.InputType;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.Toast;

public class Signin extends Activity{

Intent i=null;

ImageView im=null;

EditText tv1,tv2,tv3,tv4;

boolean flag=false;

SQLiteDatabase db=null;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.signin);

im=(ImageView)findViewById(R.id.show\_hide);

tv1=(EditText)findViewById(R.id.name);

tv2=(EditText)findViewById(R.id.email\_id);

tv3=(EditText)findViewById(R.id.phone);

tv4=(EditText)findViewById(R.id.password);

db=openOrCreateDatabase("mydb", MODE\_PRIVATE, null);

db.execSQL("create table if not exists login(name varchar,mobile\_no varchar,email\_id varchar,password varchar,flag varchar)");

im.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View arg0) {

if(flag==false)

{

im.setImageResource(R.drawable.hide);

tv4.setInputType(InputType.TYPE\_TEXT\_VARIATION\_PASSWORD);

flag=true;

}

else

{

im.setImageResource(R.drawable.show);

tv4.setInputType(129);

flag=false;

}

}

});

}

public void action(View v)

{

switch(v.getId())

{

case R.id.login:

i=new Intent(this,Login.class);

startActivityForResult(i, 500);

overridePendingTransition(R.anim.slide\_in\_top, R.anim.slide\_out\_bottom);

finish();

break;

case R.id.signin:

String name=tv1.getText().toString();

String email\_id=tv2.getText().toString();

String mobile\_no=tv3.getText().toString();

String password=tv4.getText().toString();

if(name==null||name==""||name.length()<3)

{

show("Please Enter Correct Name.");

}

else if(mobile\_no==null||mobile\_no==""||mobile\_no.length()<10)

{

show("Please Enter Correct mobile number.");

}

else if(email\_id==null||email\_id==""||email\_id.length()<10)

{

show("Please Enter Correct Email id.");

}

else if(password==null||password==""||password.length()<6)

{

show("Please Enter Strong Password.");

}

else

{

db.execSQL("insert into login values('"+name+"','"+mobile\_no+"','"+email\_id+"','"+password+"','nothing')");

i=new Intent(this,Welcome.class);

startActivityForResult(i, 500);

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);

db.close();

finish();

}

break;

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

}

public void show(String str)

{

Toast.makeText(this, str, Toast.LENGTH\_LONG).show();

}

}

**3.MainActivity.java**

package com.example.PetrolFinder;

import com.example.PetrolFinder.R;

import android.os.Bundle;

import android.view.View;

import android.app.Activity;

import android.content.Intent;

public class MainActivity extends Activity {

Intent i=null;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void login\_sigin(View v)

{

switch(v.getId())

{

case R.id.log\_in:

i=new Intent(this,Login.class);

startActivityForResult(i, 500);

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);

break;

case R.id.sign\_in:

i=new Intent(this,Signin.class);

startActivityForResult(i, 500);

overridePendingTransition(R.anim.slide\_in\_right, R.anim.slide\_out\_left);;

break;

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

overridePendingTransition(R.anim.slide\_in\_left, R.anim.slide\_out\_right);

}

}

**4.Welcome.java**

package com.example.PetrolFinder;

import com.example.PetrolFinder.R;

import android.net.Uri;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

public class Welcome extends Activity {

Button button,btn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.welcome);

addListenerOnButton();

}

public void addListenerOnButton() {

final Context context = this;

button = (Button) findViewById(R.id.button1);

btn=(Button)findViewById(R.id.button2);

button.setOnClickListener(new OnClickListener() {

public void onClick(View arg0) {

Intent intent = new Intent(context,GooglePlacesActivity.class);

startActivity(intent);

}

});

btn.setOnClickListener(new OnClickListener() {

public void onClick(View arg0){

Uri uri = Uri.parse("http://www.petrolprices.com/fuel-tax-calculator.html/");

Intent intent = new Intent(Intent.ACTION\_VIEW, uri);

startActivity(intent);

}

});

}

}

**5.GooglePlacesActivity.java**

package com.example.PetrolFinder;

import android.location.Criteria;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.support.v4.app.FragmentActivity;

import android.util.Log;

import com.example.PetrolFinder.R;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.GooglePlayServicesUtil;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

public class GooglePlacesActivity extends FragmentActivity implements LocationListener

{

private static final String GOOGLE\_API\_KEY="AIzaSyB61cWG3yFdb3sMPM5Vt3iJRSdjDg58KKw";

GoogleMap googleMap;

// EditText placeText;

double latitude = 0;

double longitude = 0;

private int PROXIMITY\_RADIUS = 5000;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

//show error dialog if GoolglePlayServices not available

if (!isGooglePlayServicesAvailable()) {

finish();

}

setContentView(R.layout.activity\_google\_places);

// placeText = (EditText) findViewById(R.id.placeText);

// Button btnFind = (Button) findViewById(R.id.btnFind);

SupportMapFragment fragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.googleMap);

googleMap = fragment.getMap();

googleMap.setMyLocationEnabled(true);

LocationManager locationManager = (LocationManager) getSystemService(LOCATION\_SERVICE);

Criteria criteria = new Criteria();

String bestProvider = locationManager.getBestProvider(criteria, true);

Location location = locationManager.getLastKnownLocation(bestProvider);

if (location != null) {

onLocationChanged(location);

}

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, this);

locationManager.requestLocationUpdates(LocationManager.NETWORK\_PROVIDER, 0, 0, this);

/\* btnFind.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

String type = placeText.getText().toString();

StringBuilder googlePlacesUrl = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?");

googlePlacesUrl.append("location=" + latitude + "," + longitude);

googlePlacesUrl.append("&radius=" + PROXIMITY\_RADIUS);

googlePlacesUrl.append("&types=" + "gas\_station");

googlePlacesUrl.append("&sensor=true");

googlePlacesUrl.append("&key=" + GOOGLE\_API\_KEY);

GooglePlacesReadTask googlePlacesReadTask = new GooglePlacesReadTask();

Object[] toPass = new Object[2];

toPass[0] = googleMap;

toPass[1] = googlePlacesUrl.toString();

Log.e("GooglePlacesActivity", googlePlacesUrl.toString());

googlePlacesReadTask.execute(toPass);

}

});\*/

}

private boolean isGooglePlayServicesAvailable() {

int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(this);

if (ConnectionResult.SUCCESS == status) {

return true;

} else {

GooglePlayServicesUtil.getErrorDialog(status, this, 0).show();

return false;

}

}

@Override

public void onLocationChanged(Location location) {

Log.e("In onLocationChanged", "location");

latitude = location.getLatitude();

longitude = location.getLongitude();

Log.e("In onLocationChanged", "location: "+ latitude +" "+ longitude);

LatLng latLng = new LatLng(latitude, longitude);

googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

googleMap.animateCamera(CameraUpdateFactory.zoomTo(12));

// String type = placeText.getText().toString();

StringBuilder googlePlacesUrl = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?");

googlePlacesUrl.append("location=" + latitude + "," + longitude);

googlePlacesUrl.append("&radius=" + PROXIMITY\_RADIUS);

googlePlacesUrl.append("&types=" + "gas\_station");

googlePlacesUrl.append("&sensor=true");

googlePlacesUrl.append("&key=" + GOOGLE\_API\_KEY);

GooglePlacesReadTask googlePlacesReadTask = new GooglePlacesReadTask(this);

Object[] toPass = new Object[2];

toPass[0] = googleMap;

toPass[1] = googlePlacesUrl.toString();

Log.e("GooglePlacesActivity", googlePlacesUrl.toString());

googlePlacesReadTask.execute(toPass);

}

@Override

public void onProviderDisabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onProviderEnabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

// TODO Auto-generated method stub

}

}

**6.** **GooglePlacesReadTask.java**

package com.example.PetrolFinder;

import android.app.Activity;

import android.os.AsyncTask;

import android.util.Log;

import com.google.android.gms.maps.GoogleMap;

public class GooglePlacesReadTask extends AsyncTask<Object, Integer, String> {

String googlePlacesData = null;

GoogleMap googleMap;

Activity activity;

public GooglePlacesReadTask(Activity activity) {

// TODO Auto-generated constructor stub

this.activity = activity;

}

@Override

protected String doInBackground(Object... inputObj) {

try {

googleMap = (GoogleMap) inputObj[0];

String googlePlacesUrl = (String) inputObj[1];

Http http = new Http();

googlePlacesData = http.read(googlePlacesUrl);

Log.e("GooglePlacesActivity", googlePlacesData);

} catch (Exception e) {

Log.d("Google Place Read Task", e.toString());

}

return googlePlacesData;

}

@Override

protected void onPostExecute(String result) {

PlacesDisplayTask placesDisplayTask = new PlacesDisplayTask(activity);

Object[] toPass = new Object[2];

toPass[0] = googleMap;

toPass[1] = result;

placesDisplayTask.execute(toPass);

}

}

**7.Http.java**

package com.example.PetrolFinder;

import android.util.Log;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class Http {

public String read(String httpUrl) throws IOException {

String httpData = "";

InputStream inputStream = null;

HttpURLConnection httpURLConnection = null;

try {

URL url = new URL(httpUrl);

httpURLConnection = (HttpURLConnection) url.openConnection();

httpURLConnection.connect();

inputStream = httpURLConnection.getInputStream();

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

StringBuffer stringBuffer = new StringBuffer();

String line = "";

while ((line = bufferedReader.readLine()) != null) {

stringBuffer.append(line);

}

httpData = stringBuffer.toString();

bufferedReader.close();

} catch (Exception e) {

Log.d("Exception - reading Http url", e.toString());

} finally {

inputStream.close();

httpURLConnection.disconnect();

}

return httpData;

}

}

**8.Places.java**

package com.example.PetrolFinder;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

public class Places {

public List<HashMap<String, String>> parse(JSONObject jsonObject)

{

JSONArray jsonArray = null;

try {

jsonArray = jsonObject.getJSONArray("results");

} catch (JSONException e)

{

e.printStackTrace();

}

return getPlaces(jsonArray);

}

private List<HashMap<String, String>> getPlaces(JSONArray jsonArray)

{

int placesCount = jsonArray.length();

List<HashMap<String, String>> placesList = new ArrayList<HashMap<String, String>>();

HashMap<String, String> placeMap = null;

for (int i = 0; i < placesCount; i++) {

try {

placeMap = getPlace((JSONObject) jsonArray.get(i));

placesList.add(placeMap);

} catch (JSONException e) {

e.printStackTrace();

}

}

return placesList;

}

private HashMap<String, String> getPlace(JSONObject googlePlaceJson)

{

HashMap<String, String> googlePlaceMap = new HashMap<String, String>();

String placeName = "-NA-";

String vicinity = "-NA-";

String latitude = "";

String longitude = "";

String reference = "";

try {

if (!googlePlaceJson.isNull("name"))

{

placeName = googlePlaceJson.getString("name");

}

if (!googlePlaceJson.isNull("vicinity"))

{

vicinity = googlePlaceJson.getString("vicinity");

}

latitude = googlePlaceJson.getJSONObject("geometry").getJSONObject("location").getString("lat");

longitude = googlePlaceJson.getJSONObject("geometry").getJSONObject("location").getString("lng");

reference = googlePlaceJson.getString("reference");

googlePlaceMap.put("place\_name", placeName);

googlePlaceMap.put("vicinity", vicinity);

googlePlaceMap.put("lat", latitude);

googlePlaceMap.put("lng", longitude);

googlePlaceMap.put("reference", reference);

} catch (JSONException e)

{

e.printStackTrace();

}

return googlePlaceMap;

}

}

**9. PlacesDisplayTask.java**

package com.example.PetrolFinder;

import android.app.Activity;

import android.content.Intent;

import android.os.AsyncTask;

import android.util.Log;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.GoogleMap.OnInfoWindowClickListener;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.Marker;

import com.google.android.gms.maps.model.MarkerOptions;

import org.json.JSONObject;

import java.util.HashMap;

import java.util.List;

public class PlacesDisplayTask extends AsyncTask<Object, Integer, List<HashMap<String, String>>> {

JSONObject googlePlacesJson;

GoogleMap googleMap;

Activity activity;

public PlacesDisplayTask(Activity activity) {

// TODO Auto-generated constructor stub

this.activity = activity;

}

@Override

protected List<HashMap<String, String>> doInBackground(Object... inputObj) {

List<HashMap<String, String>> googlePlacesList = null;

Places placeJsonParser = new Places();

try {

googleMap = (GoogleMap) inputObj[0];

googlePlacesJson = new JSONObject((String) inputObj[1]);

googlePlacesList = placeJsonParser.parse(googlePlacesJson);

Log.e("GooglePlacesActivity", googlePlacesList.size() + " ");

} catch (Exception e) {

Log.d("Exception", e.toString());

}

return googlePlacesList;

}

@Override

protected void onPostExecute(List<HashMap<String, String>> list) {

googleMap.clear();

for (int i = 0; i < list.size(); i++) {

MarkerOptions markerOptions = new MarkerOptions();

HashMap<String, String> googlePlace = list.get(i);

double lat = Double.parseDouble(googlePlace.get("lat"));

double lng = Double.parseDouble(googlePlace.get("lng"));

final String placeName = googlePlace.get("place\_name");

final String vicinity = googlePlace.get("vicinity");

LatLng latLng = new LatLng(lat, lng);

markerOptions.position(latLng);

markerOptions.title(placeName + " : " + vicinity);

googleMap.setOnInfoWindowClickListener(new OnInfoWindowClickListener() {

@Override

public void onInfoWindowClick(Marker arg0) {

// TODO Auto-generated method stub

Intent intent = new Intent(activity, Token.class);

intent.putExtra("data"," "+arg0.getTitle() );

activity.startActivity(intent);

}

});

googleMap.addMarker(markerOptions);

}

}

}

**10. Token.java**

package com.example.PetrolFinder;

import com.example.PetrolFinder.R;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.widget.TextView;

public class Token extends Activity {

TextView txtView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.token);

txtView = (TextView) findViewById(R.id.textView1);

Intent intent = getIntent();

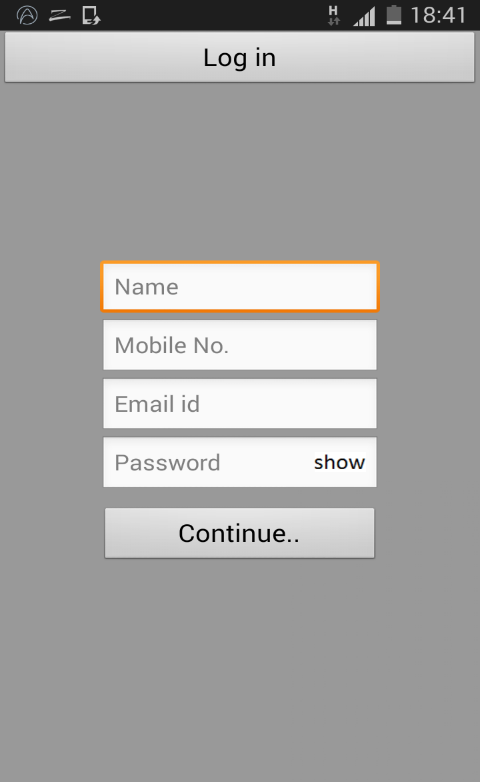
String data = intent.getStringExtra("data");

txtView.setText("Thank you for selecting this station"+data);}}

**Screenshots:**



1)This is the welcome page for the petrol finder app that has sign in and login buttons for user registration.



2)This is the sign in page wherein user has to register first to enter the app.



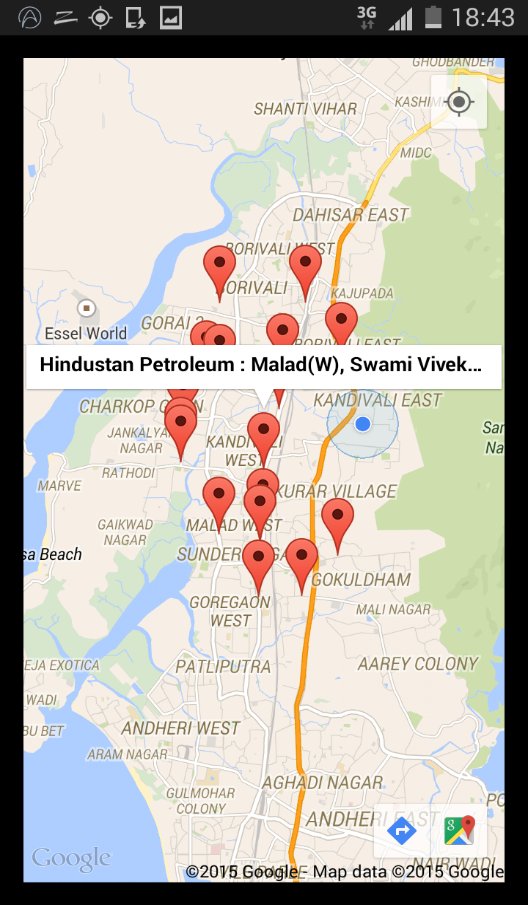
3 )Now here is the login page. If you are a registered user just type in your mobile number and password to enter the application.



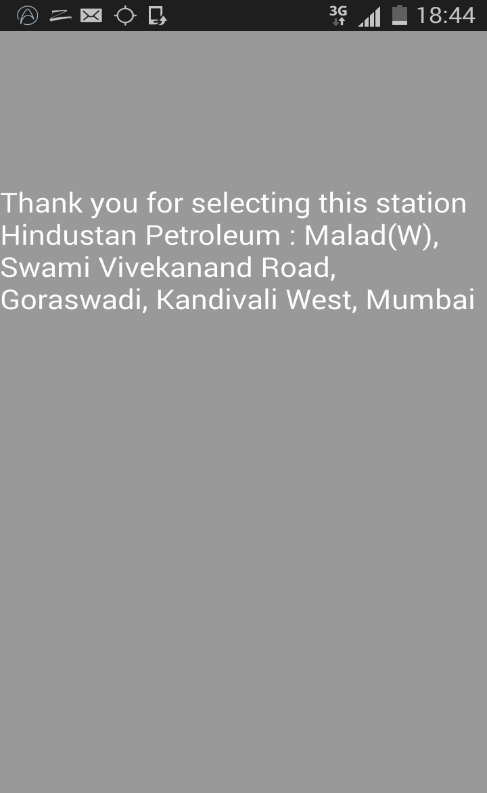
4)If user enters wrong details there will be an intimation regarding it and the “show” and “hide” text helps the user to confirm his/her password.



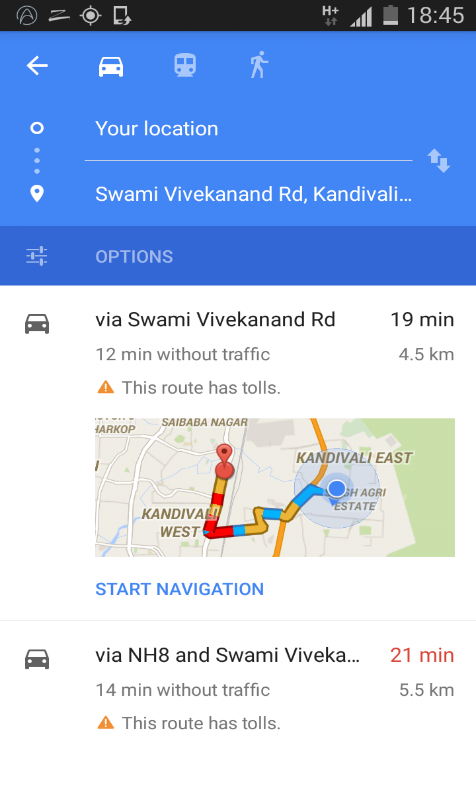
5) Here is the main menu of the app.



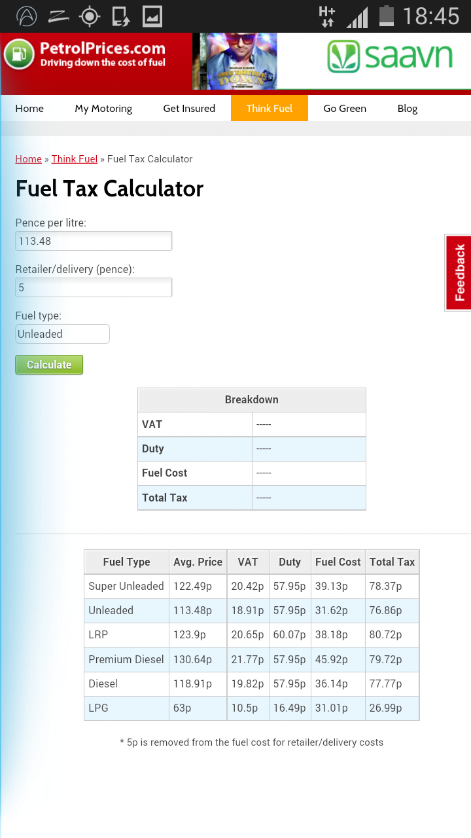
6) Once the user clicks on petrol pump locator the application tracks the user location and displays all the petrol stations near his location. This is possible due to the gps tracker and network provider.



7)If the user clicks on the white band shown on the map which points at a particular petrol station he/she will be redirected to the next activity displaying the entire address of the location.



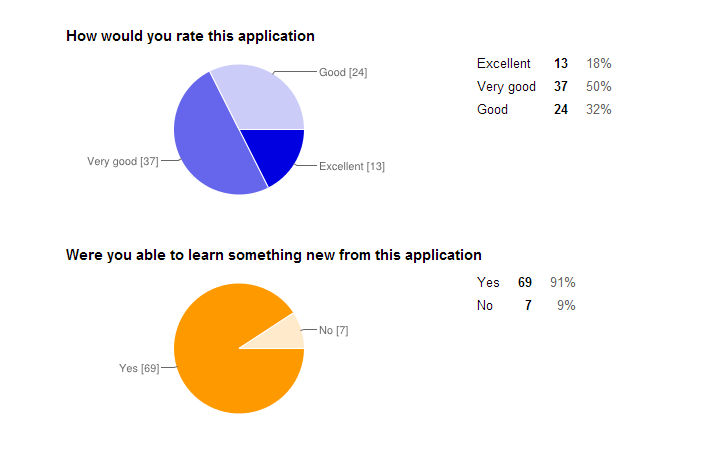
8) Not only this the application is also connected to the google maps which shows the user navigation from current location to the petrol pump whichever the user selects.



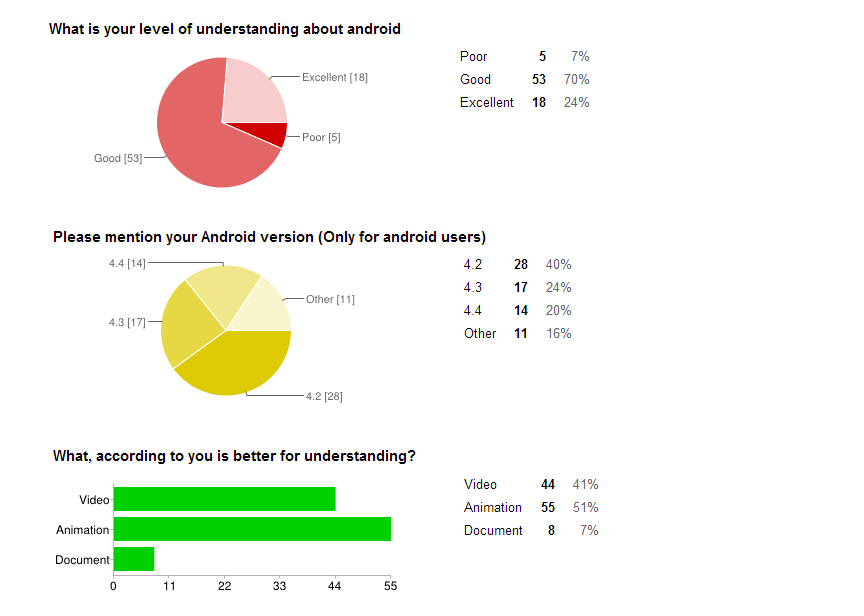
9) The fuel tax calculator button redirects the user to a webpage which shows him the current cost of fuel and also the VAT. The fuel tax calculator is useful for retailers and distributors to keep a track about the tax information.

**Chapter 5**

**Testing Result**

****

**Figure 5.1: Overall Testing Result of Application [1]**

****

**Figure 5.2: Overall Testing Result of Application [2]**

**Conclusion**

Smart phones and Tablets are more common than computers today. Almost everyone in the World make regular use of smart phones in their day to day lives. People can get a lot of Different benefits from smart phones and that too in a very portable and mobile manner.

The android operating system presents never before seen flexibility and support for third-Party applications. This has given rise to a huge amount of popularity of the Android OS not only among consumers, but also among developers. Android application development is fast becoming a separate field of information technology. More and more independent app Developers and app development companies are taking interest in this OS and are coming up with some of the best apps around. Hence, one can understand the importance of android application development.

Initially mobile phones were developed only for voice communication but now days the scenario has changed, voice communication is just one aspect of a mobile phone. There are other aspects which are major focus of interest. Two such major factors are web browser and GPS services. Both of these functionalities are already implemented but are only in the hands of manufacturers not in the hands of users because of proprietary issues, the system does not allow the user to access the mobile hardware directly. But now, after the release of android based open source mobile phone a user can access the hardware directly and design customized native applications to develop Web and GPS enabled services and can program the other hardware components like camera etc. Building an android application on Supply chain management for oil industry will reduce the nature of complexity which persists between the supplier and the customer since ultimately “Customer is a King”. The transportation cost,lead time, everything should be handled appropriately to eliminate any inflexibilites in this chain.The customer should get timely access to all the resources he needs.Hence,this app will fulfill those requirements.

Supply Chain Management of Oil & Gas industry is of the most volatile & challenging nature. Hence, a proper study is required which can be used as a reference to know about the past and current practices of the country in this sector along with the issues faced, in order to be able to devise more options when planning for future.

The purpose is to:

* Study the current and past supply chain practices in the Oil & Gas industry.
* Understand the current issues and challenges of the industry.
* Find possible ways to optimize the value chain of the industry.

**Future Scope:**

Technology is transforming and rejuvenating at a fast pace within no time Android has become a favorite of millions of mobile developers and users. It is an open source Linux based operating system designed specifically for Smartphone’s and tablets. In such a short span of time, the growth of android market has been fabulous. The ease of use coupled with convenience and entertainment factor have been the powerful features of this technology.

Future of android is beyond imagination. It has opened a new stream of technological advancements. And interestingly, there seems no end to experimenting and evolving new applications and mobile phones using this innovative and dynamic technology. In years to come the usage of this technology is bound to increase because of its flexibility to develop applications of different kinds. There has been an unexpected rise in the demands of [Android developers](http://www.moonmicrosystem.co.in/android-app-development.php) and in future this demand is expected to increase with time. With each year passing, we are expected to have more and more developers joining the pool of talent because as it is an open source platform, the scope to create something new is infinite. It totally depends on the imagination and skills of the developers that what they wish to create out of this amazing technology.

* This Application can be extended to show supply chain complexities to users by linking it to videos.
* This Application can be extended at an industrial level by connecting it to servers of the petrol stations and getting real time information about their inventory system and other such details.
* This application can also be possible to implement for iphone, symbian and windows user.
* We can also provide more material and information offline to user according to their necessary by increasing storage space.

**References:**

1. Supply chain Practices & Optimization in Petroleum Industry

By:Waqis Ali Tunio,MS(Manufacturing Engineering & Management)

National University Of Sciences & Technology (NUST)-Pakistan

International Journal paper

1. Managing And Mitigating the Upstream Petroleum Industry Supply Chain Risks:Leveraging Analytic Hierarchy Process,By:Charles A.Briggs

Southern University at New Orleans

1. Gainsborough, M. (2006). Building World-class supply Chain Capability in the Downstream Oil Business. *Business Briefing: Oil and Gas Processing Review*. 29-32.
2. Hasini, E. (2008). Supply Chain Optimization: Current Practices and Overview of Emerging Research Opportunities. *Information System and Operation Research. 42*(2), 93-96.
3. Hölshler, J., Bachan, R. & Stimpson, A. (2008). Oil Demand in China: an Econometric Approach. *International Journal of Emerging Markets. 3*(1), 54-70.
4. Hussain, R., Assavapokee, T. & Khumawala, B. (2006). Supply Chain Management in the Petroleum Industry: Challenges and Opportunities. *International Journal of Global Logistics & Supply Chain Management. 1*(2), 90-97. Jasuja, A., Sowmya, A., Chaudhary, A., Kanade S. & Panda.

**Web References:**

* <http://www.tutorialspoint.com/android/>
* <https://developer.android.com/training/index.html>
* <http://stackoverflow.com/questions/13194081/how-to-open-a-second-activity-on-click-of-button-in-android-app>
* <http://stackoverflow.com/questions/13660758/add-google-api-to-existing-eclipse-project>

Paper Published:

* Rachana Mahale, Pranoti Kulkarni & Kanika Danke Published paper on ' Android application on Supply chain management in oil industry' in **Multicon 2015** held at TCET, Mumbai.